# Introducción a la Criptografía Moderna
# Presentación del Curso

**Rodrigo Abarzúa**[†],

[†] Universidad de Santiago de Chile
rodrigo.abarzua@usach.cl

3 de abril de 2014

# Objetivo del Curso

El objetivo de este curso es introducir a los alumnos en los aspectos teóricos-aplicados de la criptografía moderna, para esto, se realizara un estudio de los principales algoritmos utilizados en la actualidad y estandarizados por NIST para los distintos objetivos buscados por criptografía actual.

# Bibliografía

Aspectos Teóricos:


1. Understanding Cryptography, A textbook for Student and Practitioners. Christof Paar and Jan Pelzl. Springer-Verlag Berlin Heidelberg 2010.
2. Cryptography Theory and Practice. Douglas R. Stinson. The CRC Press Series on Discrete Mathematics and Its Applications.
3. Cryptography: An Introducction 3rd Edition, Nigel Smart, Department of Computer Science, University of Bristol Este libro puede ser bajado de la dirección: *http* : *//www.cs.bris.ac.uk/ nigel/Crypto_Book/*
4. An Introduction to Mathematical Cryptography, Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. Undergraduate Texts in Mathematics, Springer-Verlag 2008.
5. Guide to Elliptic Curve Cryptography, Darrel Hankerson, Alfred Menezes and Scott Vanstone. Springer-Verlag 2004.
6. Handbook of Elliptic and Hyperelliptic Curve Cryptography, Roberto Avanzi, et al. Discrete Mathematics and Its Applications, Champman & Hall/CRC, 2005.

# Bibliografía

Aspectos Aplicados:


1. Applied Cryptography, Protocols, Algorithms and Source Code in C. Bruce Schneier, Jhon Whiley Sons, 1996.
2. Cryptography in C and C++, Second Edition Michael Welschenbach, Apress 2005
3. Segure Programming Cookbook for C and C++, Jhon Viega & Matt Messier, O'Reilly 2003

# Evaluación de Curso

- El curso se evaluará en base a dos pruebas específicas programadas (PEP) con igual ponderación. Para aprobar la cátedra nota final $\geq 4{,}0$
- En ejercicios de laboratorio se solicitara a los alumnos que implemente en C algunos de los algoritmos expuestos en aula. Para aprobar laboratorio nota final $\geq 4{,}0$
- Los alumnos deberán estudiar y presentar algún articulo de investigación de relevancia para el curso.
- Se solicitara como mínimo un 75 % de asistencia a clases.

NOTA F.= 0.6*(Promedio Pruebas)+0.2*(Exposición de Articulo)+0.2*(Laboratorio)

# Objetivos de la Criptografía

La criptografía es un campo multidisciplinario, que abarca:

1. Matemáticas: álgebra, los grupos finitos, anillos y cuerpos.
2. Ingeniería Eléctrica: diseño de hardware, ASIC, FPGA.
3. Ciencias de la Computación: algoritmos, teoría de la complejidad, diseño de software, sistemas integrados (embedded systems).

# Objetivos de la Criptografía

Supongamos que dos personas Alice (A) y Bob (B) se desean comunicar en un canal inseguro, por ejemplo, este canal podría ser Internet o una Red LAN, o una comunicación inalámbrica a través de un smartphone o celular. El problema nace cuando una persona no autorizada digamos Eve (E) es capaz de escuchar o leer la comunicación que están realizando A y B, a este tipo de escuchadores no autorizados es conocido como *eavesdropping*.

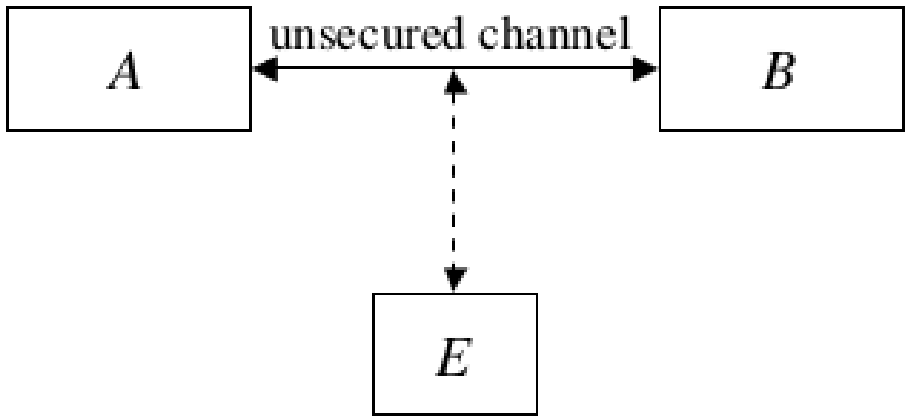



Figura: Modelo Básico de Comunicación

# Objetivos de la Criptografía

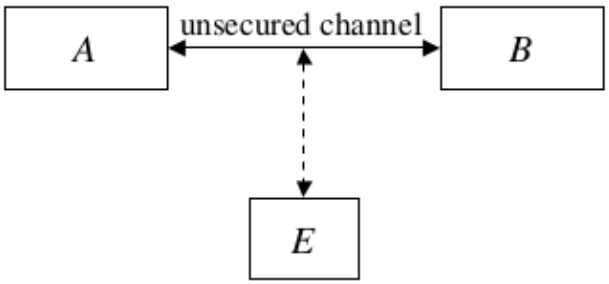



Figura: Modelo Básico de Comunicación

Un primer objetivo: es que Alice y Bob desean la Confidenciabilidad de la comunicación, por ejemplo, si Alice y Bob están intercambiando información delicada, como lo son claves de acceso a lugares restringidos, o estrategias y proyectos de negocios altamente competitivos, o tecnologías de punta en las distintas áreas de la producción, etc.

# Objetivos de la Criptografía

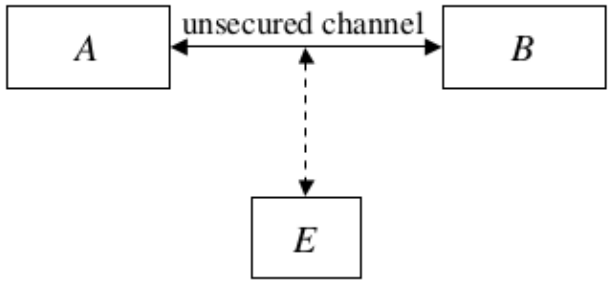



Figura: Modelo Básico de Comunicación

Un segundo objetivo que podría ser de interés para Alice y Bob es la Integridad de la información, es decir, asegurar que la información intercambiada no sea alterada por Eve y que Bob o Alice pueda observar si un mensaje recibido ha sido alterado por Eve.

# Objetivos de la Criptografía

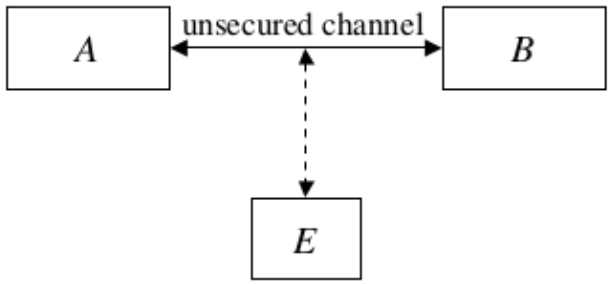



Figura: Modelo Básico de Comunicación

Un tercer objetivo podría ser la Autentificación de Origen de la Información, es decir que Bob este seguro que es Alice es quien le envío la información.

# Objetivos de la Criptografía

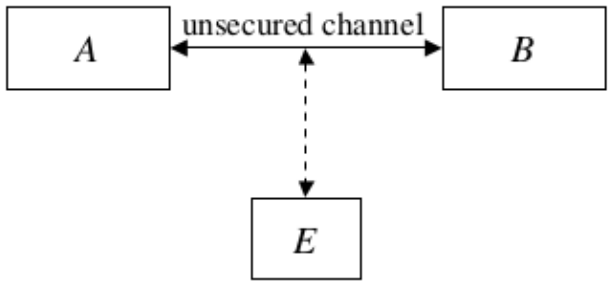



Figura: Modelo Básico de Comunicación

Un cuarto objetivo podría ser la Autentificación de la Identidad, esto quiere decir, que Bob es capaz de corroborar la identidad de Alice.

# Objetivos de la Criptografía

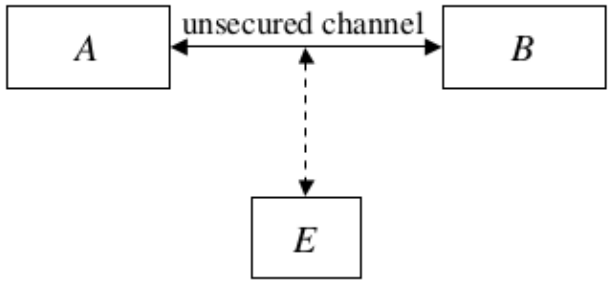



Figura: Modelo Básico de Comunicación

Un quinto objetivo podría ser el No repudio de la Información, es decir, que si Alice envío una información a Bob, luego, Alice no pueda negar que fue ella quien envío dicha información. Existen otros objetivos en la criptografía como el control de acceso, el anonimato, ect.

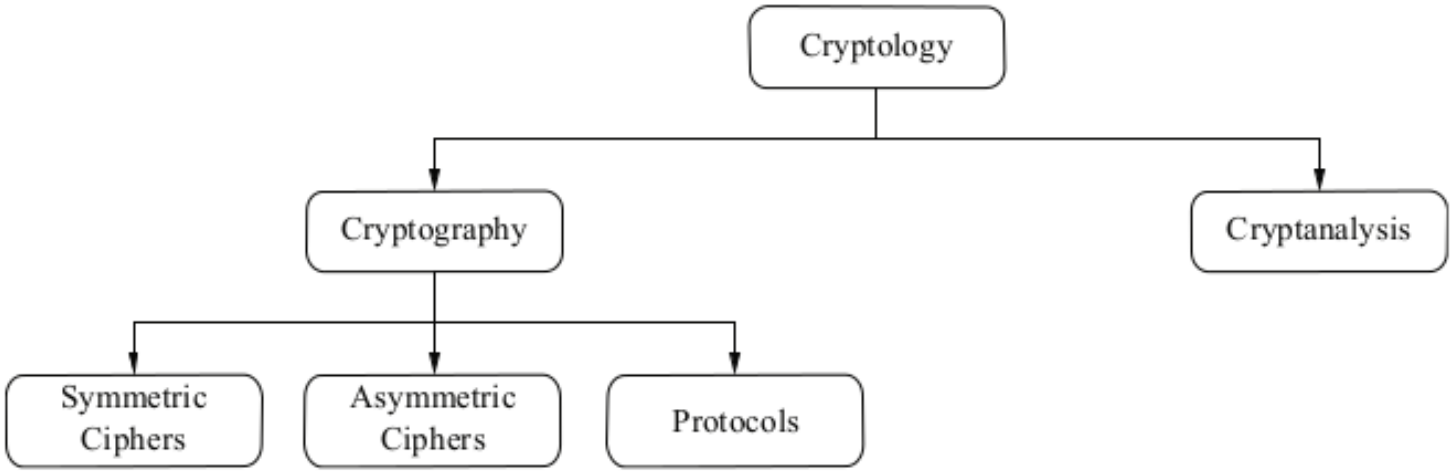



Figura: Overview del área de la Criptología

- La Criptología se divide en dos áreas, la Criptografía y el Criptoanálisis, la primera es la ciencia de escribir información de manera secreta, escondiendo el mensaje original a cualquier extraño.
- La segunda área, el Criptoanálisis es la ciencia y el arte de quebrar los criptosistemas.

En este curso nos focalizaremos en el área de la Criptografía, que se puede dividir en tres principales áreas: Los esquemas de *Criptografía Simétrica*, los esquemas de *Criptografía Asimétrica* y algunos *Protocolos*.

# Criptografía Simétrica

La Criptografía Simétrica o *Symmetric key shemes* se basa en el hecho que la información $x$ que Alice envíe a Bob es manipulada por algún algoritmos criptográfico (este algoritmo depende de que tipo de objetivo se busca, Confidenciabilidad, Autentificación, etc.) con una clave $k$ que Alice y Bob "comparten". Algunos esquemas de esta familia de criptosistemas que estudiaremos en el curso para alcanzar confidenciabilidad son: Data Encryption Standard (DES) y el Advanced Encryption Standard (AES). Para la Autentificación, estudiaremos Algoritmos Message Autentification Code (MAC) y (HMAC)

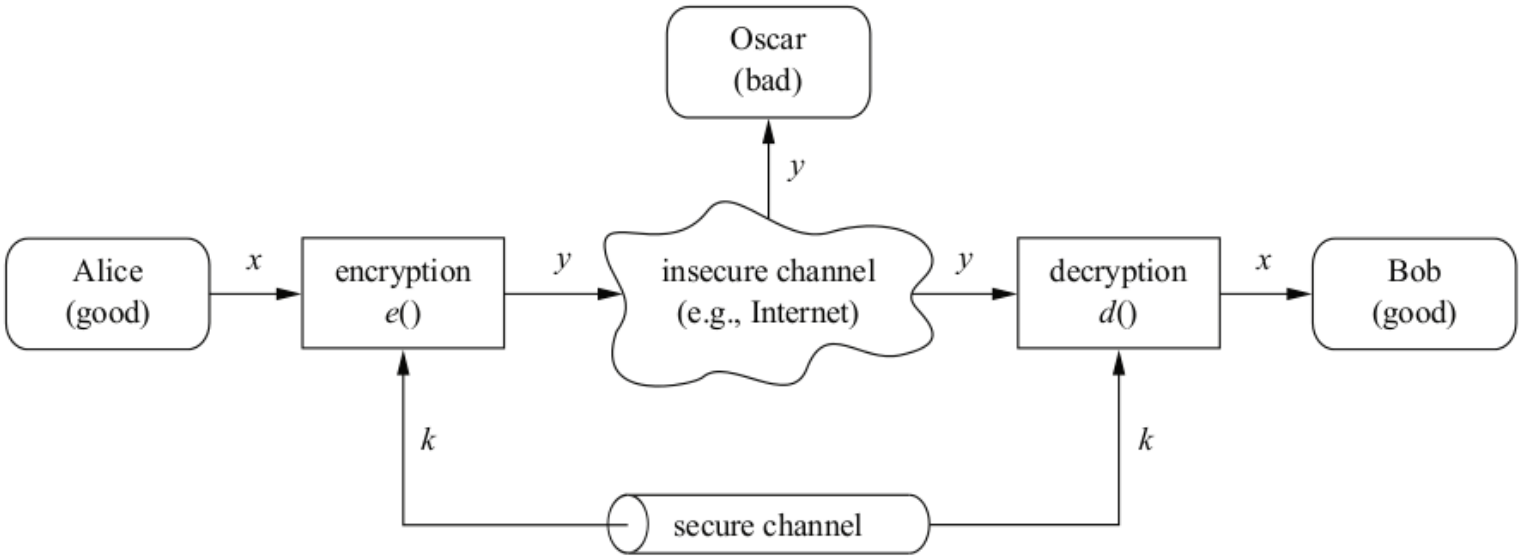



Figura: Symmetric-key Cryptosystem

# Criptografía Simétrica

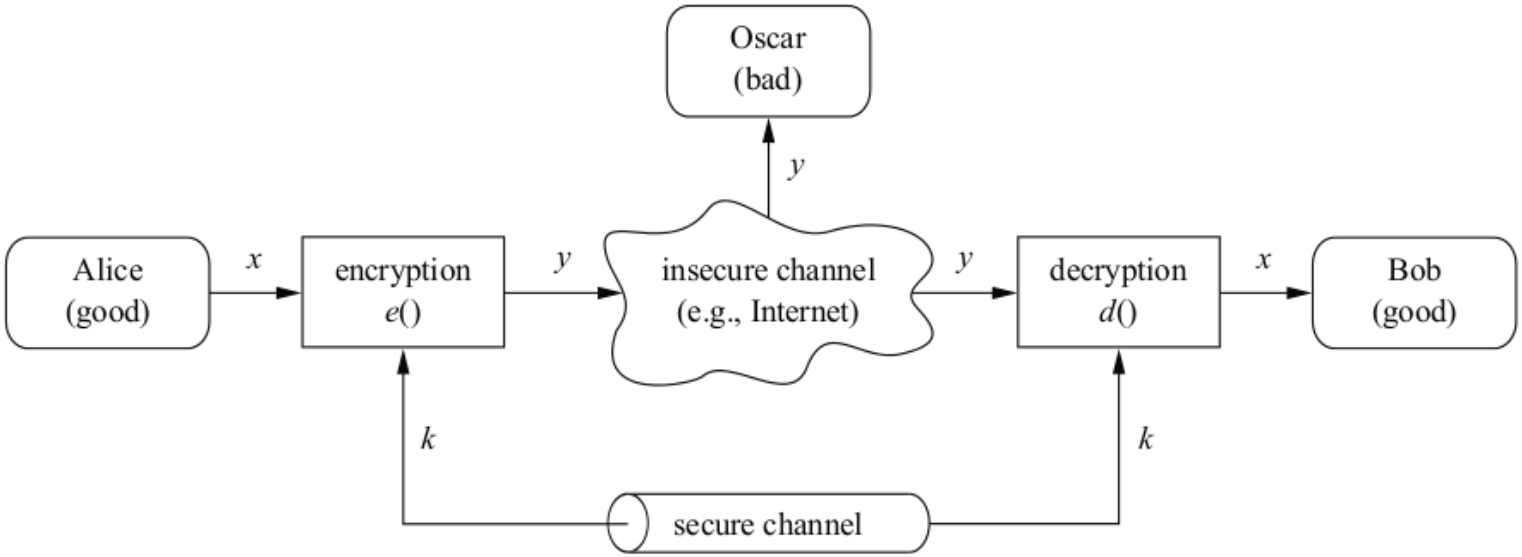



Figura: Symmetric-key Cryptosystem

Las variables $x$, $y$ y $k$ de la figura tienen nombres especiales en criptografía:

- $x$ es llamado el *plaintext* or algunas veces *cleartext*,
- $y$ es llamado *ciphertext*,
- $k$ es llamado la *clave*,
- el conjunto de todas las posibles claves es llamado el *espacio de claves*.

# Criptografía Simétrica

## Observación

- *Es importante observar que en criptografía el único secreto que se debe mantener entre dos entidades es la clave $k$, es decir, los algoritmos criptográficos deben ser públicos, ya que, de esta manera son testados y probados su fortaleza y así demostrar que son sistemas difíciles de quebrar.*

## Observación

- *La mayor ventaja de los criptosistemas simétricos es su alta eficiencia ya que su computo son rápidos.*

## Observación

- *Dado que Alice y Bob utilizan la misma clave secreta $k$, el inconveniente de estos criptosistemas es el llamado problema de distribución de claves o key distribution problem, ya que esta distribución se debe realizar de manera que la clave $k$ permanezca secreta y autenticada. Para solucionar este problema se puede usar un canal físico seguro o un servicio que entregue una tercera parte del sistema el cual elija la clave secreta $k$ y entregue físicamente a Alice y Bob.*

## Observación

- *Otra desventaja de estos criptosistemas es el llamado problema del manejo de clave o key management problem. En una red con $N$ entidades, cada entidad debe mantener $N-1$ claves de cada una de las $N-1$ entidades. Esta situación es resuelta utilizando una tercera entidad conocido en criptografía como servicio de un on-line trusted third-party que distribuye las claves requeridas para cada seción en donde dos entidades en la red deseen utilizar algún servicio criptográfico.*

# Data Encryption Standard, DES

1. EL *Data Encryption Standard (DES)* es el mas popular algoritmo de cifrado de bloques.
2. Aun cuando es considerado inseguro para algunos ataques, "dado que el conjunto de claves es muy pequeño", se sigue utilizando en algunas aplicaciones.
3. Además, en la actualidad se considera que al aplicar 3 veces el algoritmos DES (conocido como 3DES o *triple DES*) resulta ser una algoritmo de cifrado muy seguro y que en la actualidad es aún utilizado ampliamente.
4. El estudio del algoritmo DES es importante ya que ha inspirado a varios algoritmos simétricos de cifrado actuales.
5. Como el algoritmo DES es un cifrador simétrico entonces utiliza la misma clave para encriptar como desencriptar.

# Data Encryption Standard, DES

1. EL *Data Encryption Standard (DES)* es el mas popular algoritmo de cifrado de bloques.
2. Aun cuando es considerado inseguro para algunos ataques, "dado que el conjunto de claves es muy pequeño", se sigue utilizando en algunas aplicaciones.
3. Además, en la actualidad se considera que al aplicar 3 veces el algoritmos DES (conocido como 3DES o *triple DES*) resulta ser una algoritmo de cifrado muy seguro y que en la actualidad es aún utilizado ampliamente.
4. El estudio del algoritmo DES es importante ya que ha inspirado a varios algoritmos simétricos de cifrado actuales.
5. Como el algoritmo DES es un cifrador simétrico entonces utiliza la misma clave para encriptar como desencriptar.

# Data Encryption Standard, DES

1. EL *Data Encryption Standard (DES)* es el mas popular algoritmo de cifrado de bloques.
2. Aun cuando es considerado inseguro para algunos ataques, "dado que el conjunto de claves es muy pequeño", se sigue utilizando en algunas aplicaciones.
3. Además, en la actualidad se considera que al aplicar 3 veces el algoritmos DES (conocido como 3DES o *triple DES*) resulta ser una algoritmo de cifrado muy seguro y que en la actualidad es aún utilizado ampliamente.
4. El estudio del algoritmo DES es importante ya que ha inspirado a varios algoritmos simétricos de cifrado actuales.
5. Como el algoritmo DES es un cifrador simétrico entonces utiliza la misma clave para encriptar como desencriptar.

# Data Encryption Standard, DES

1. EL *Data Encryption Standard (DES)* es el mas popular algoritmo de cifrado de bloques.
2. Aun cuando es considerado inseguro para algunos ataques, "dado que el conjunto de claves es muy pequeño", se sigue utilizando en algunas aplicaciones.
3. Además, en la actualidad se considera que al aplicar 3 veces el algoritmos DES (conocido como 3DES o *triple DES*) resulta ser una algoritmo de cifrado muy seguro y que en la actualidad es aún utilizado ampliamente.
4. El estudio del algoritmo DES es importante ya que ha inspirado a varios algoritmos simétricos de cifrado actuales.
5. Como el algoritmo DES es un cifrador simétrico entonces utiliza la misma clave para encriptar como desencriptar.

# Data Encryption Standard, DES

1. EL *Data Encryption Standard (DES)* es el mas popular algoritmo de cifrado de bloques.
2. Aun cuando es considerado inseguro para algunos ataques, "dado que el conjunto de claves es muy pequeño", se sigue utilizando en algunas aplicaciones.
3. Además, en la actualidad se considera que al aplicar 3 veces el algoritmos DES (conocido como 3DES o *triple DES*) resulta ser una algoritmo de cifrado muy seguro y que en la actualidad es aún utilizado ampliamente.
4. El estudio del algoritmo DES es importante ya que ha inspirado a varios algoritmos simétricos de cifrado actuales.
5. Como el algoritmo DES es un cifrador simétrico entonces utiliza la misma clave para encriptar como desencriptar.

## Definición

*Dado $S$ un conjunto finito de elementos. Una permutación $p : S \to S$ es una función biyectiva.*

## Ejemplo

*Sea $S = \{1, 2, 3, 4, 5\}$. Una permutación $p : S \to S$ es una biyección definida como:*

$$p(1) = 3, p(2) = 5, p(3) = 4, p(5) = 1.$$

*Su representación como arreglo es:*

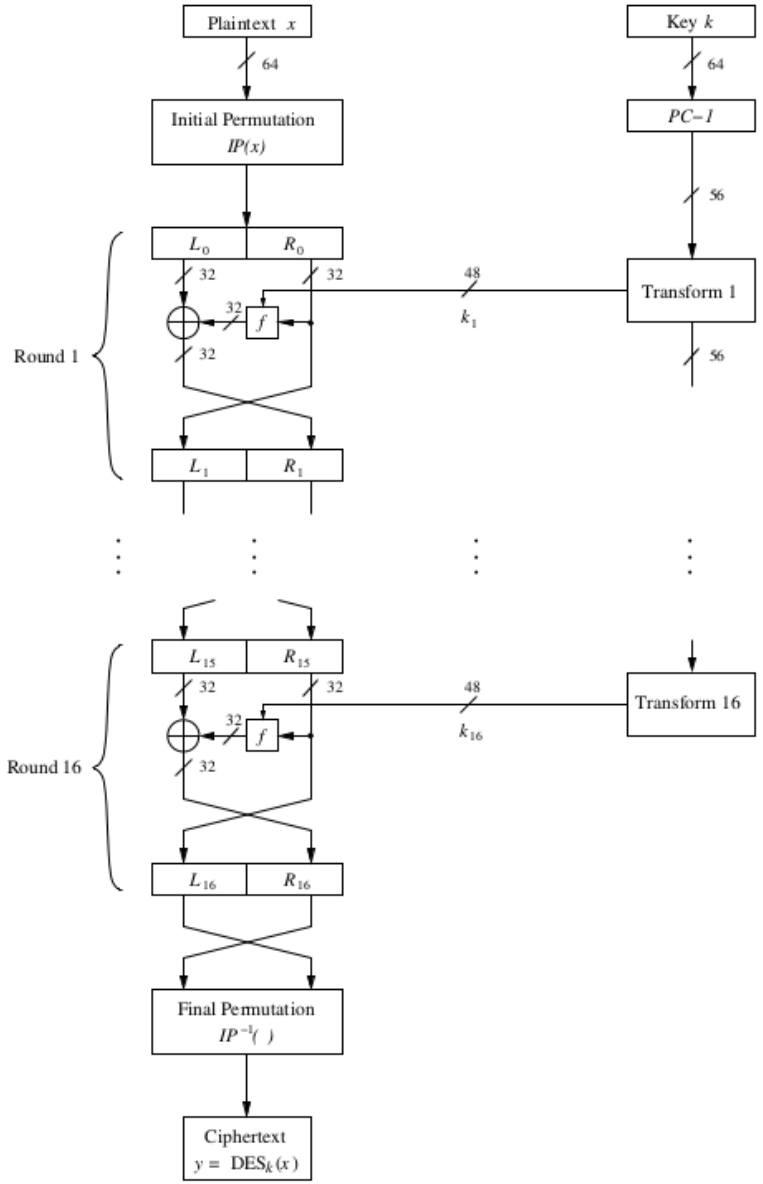
$$p = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 5 & 4 & 2 & 1 \end{bmatrix}$$

*Como p es biyectiva, entonces tiene inversa,*

$$p^{-1} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 1 & 3 & 2 \end{bmatrix}$$

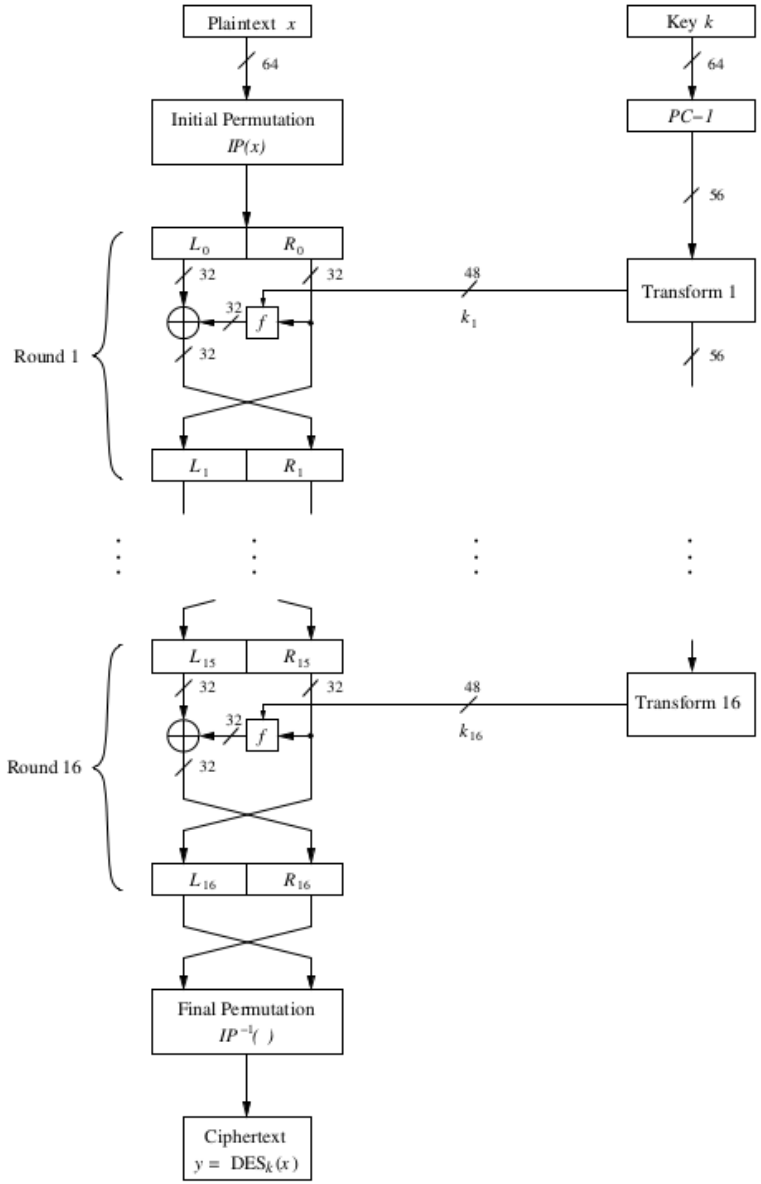
# Data Encryption Standard, DES

- Al algoritmo DES opera con bloques de plaintext $x$ de 64−bits con una clave de $k$ de 56−bits y la salida del algoritmo, es decir, el cibertext $y$ de 64-bits.
- Después de una permutación inicial denotada por $IP$ aplicada al plaintext $x$ ($IP(x)$) este bloque ($IP(x)$) es dividido en dos bloques de 32−bits cada uno, la mitad izquierda $L_i$ y la mitad derecha $R_i$. El algoritmo DES opera 16 ejecuciones idénticas utilizando una función $f$ (que describiremos más adelante), esta función mezcla los bloques del lado derecho $R_i$ con subclaves $K_i$ que son derivadas de la clave secreta $K$.



# Data Encryption Standard, DES

- Al algoritmo DES opera con bloques de plaintext $x$ de 64−bits con una clave de $k$ de 56−bits y la salida del algoritmo, es decir, el cibertext $y$ de 64-bits.
- Después de una permutación inicial denotada por $IP$ aplicada al plaintext $x$ ($IP(x)$) este bloque ($IP(x)$) es dividido en dos bloques de 32−bits cada uno, la mitad izquierda $L_i$ y la mitad derecha $R_i$. El algoritmo DES opera 16 ejecuciones idénticas utilizando una función $f$ (que describiremos más adelante), esta función mezcla los bloques del lado derecho $R_i$ con subclaves $K_i$ que son derivadas de la clave secreta $K$.



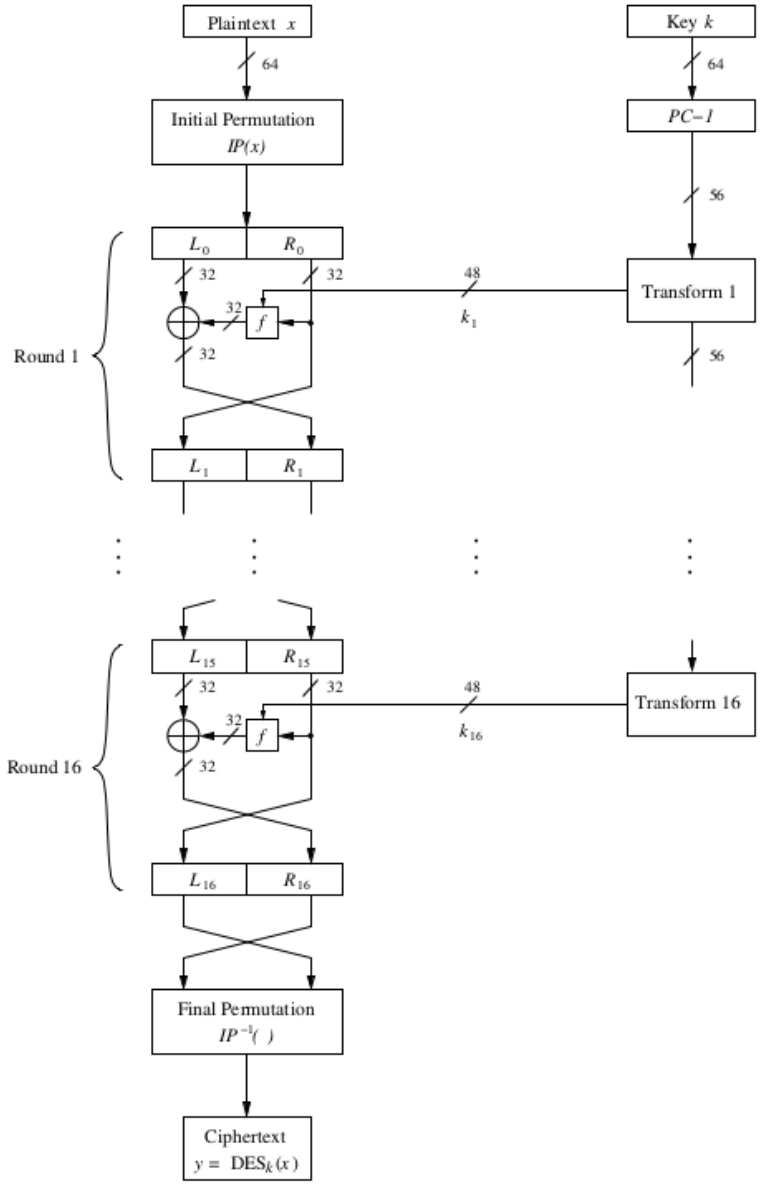
# Data Encryption Standard, DES

- Estas ejecuciones se pueden expresar como:

$$L_i = R_i$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

para $i = 0, \ldots, 16$. Después de las 16 ejecuciones, el bloque del lado derecho $R_{16}$ y del izquierdo $L_{16}$ son mezclados con una permutación inversa $IP^{-1}$ finalizando el algoritmo.

- Para cada ronda la función $f$ utiliza subclaves $K_i$ derivadas de la clave principal $K$ de 56-bits usando el llamado clave "schedule".



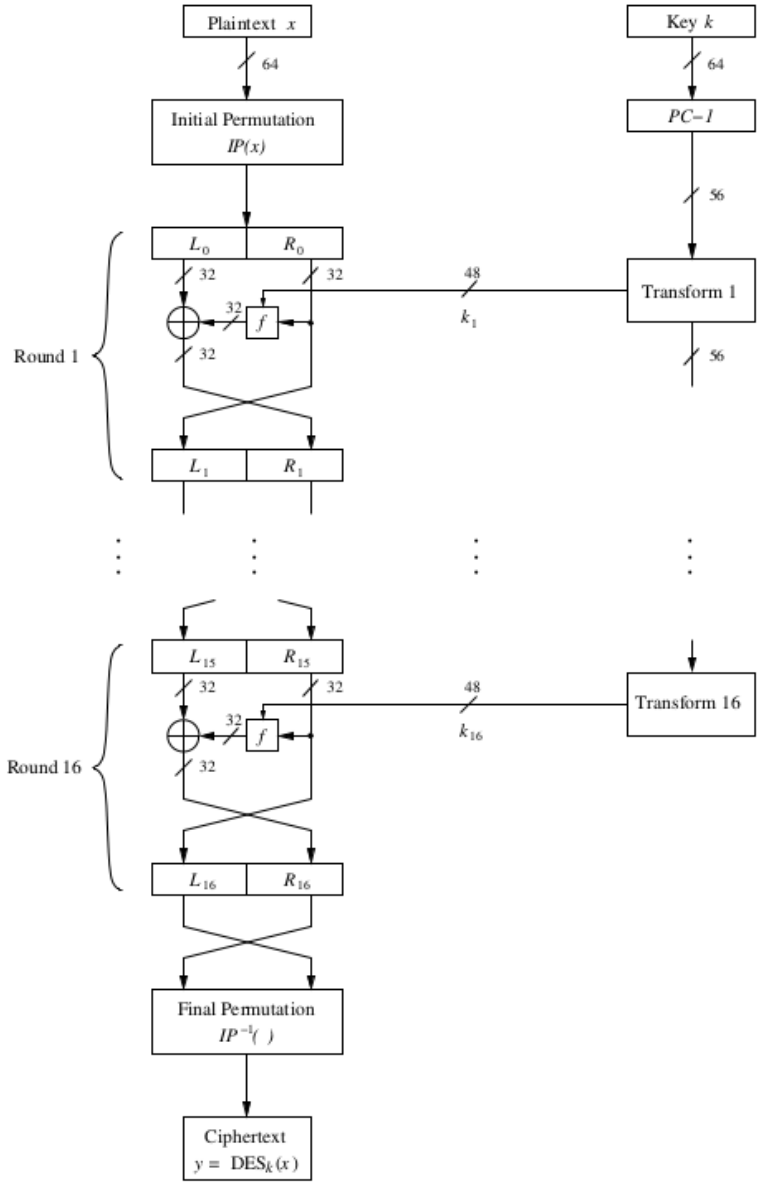
# Data Encryption Standard, DES

- Estas ejecuciones se pueden expresar como:

$$L_i = R_i$$
$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i)$$

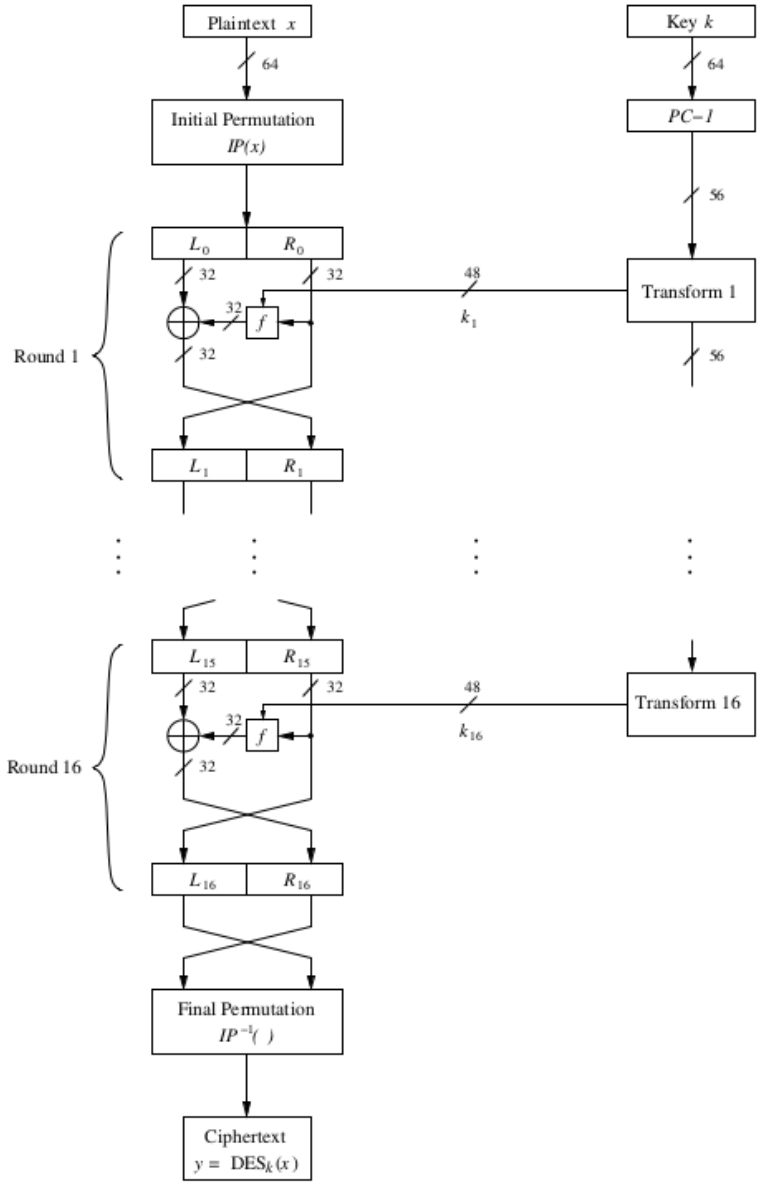
para $i = 0, \ldots, 16$. Después de las 16 ejecuciones, el bloque del lado derecho $R_{16}$ y del izquierdo $L_{16}$ son mezclados con una permutación inversa $IP^{-1}$ finalizando el algoritmo.

- Para cada ronda la función $f$ utiliza subclaves $K_i$ derivadas de la clave principal $K$ de 56-bits usando el llamado clave "schedule".



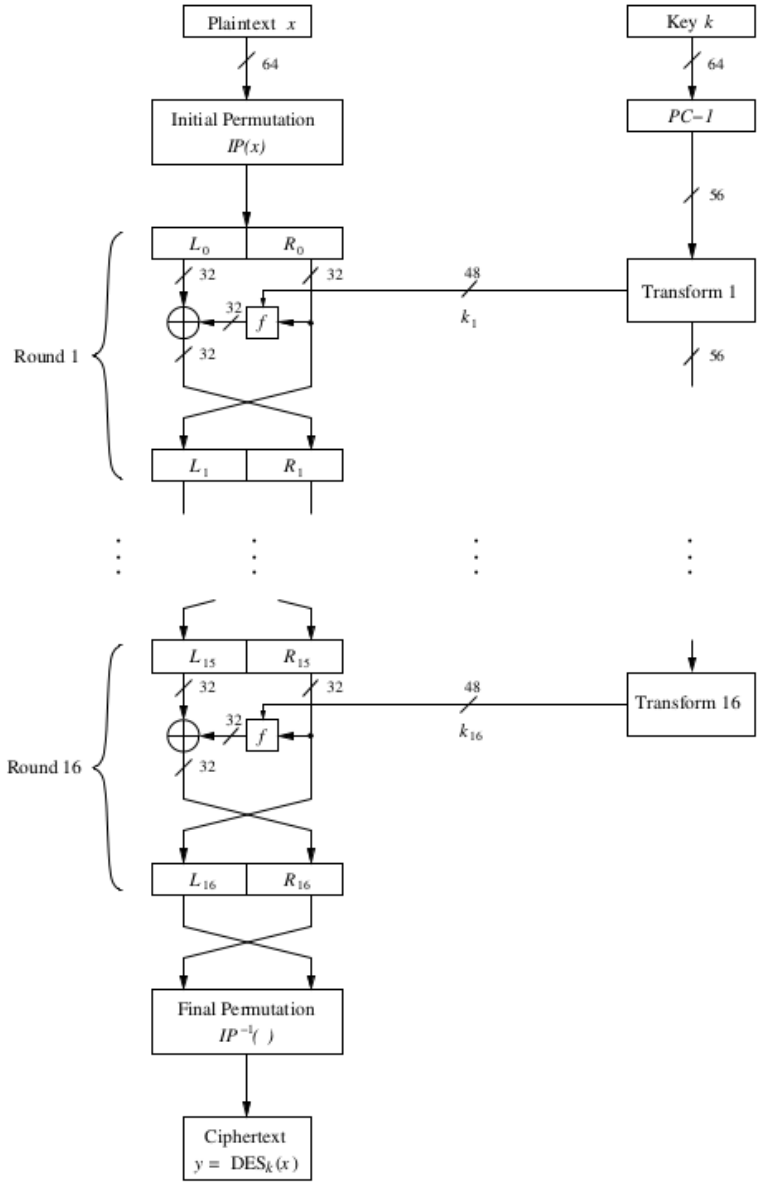
# Estructura Interna del DES

- La estructura del algoritmo DES se describe en la Figura 9.
- En lo que sigue explicaremos el funcionamiento de una ronda del algoritmo DES, (ya que se repiten 16 veces), la permutación inicial $IP$ y su inversa $IP^{-1}$, la función $f$, y la clave "schedule".



# Estructura Interna del DES

- La estructura del algoritmo DES se describe en la Figura 9.
- En lo que sigue explicaremos el funcionamiento de una ronda del algoritmo DES, (ya que se repiten 16 veces), la permutación inicial $IP$ y su inversa $IP^{-1}$, la función $f$, y la clave "schedule".

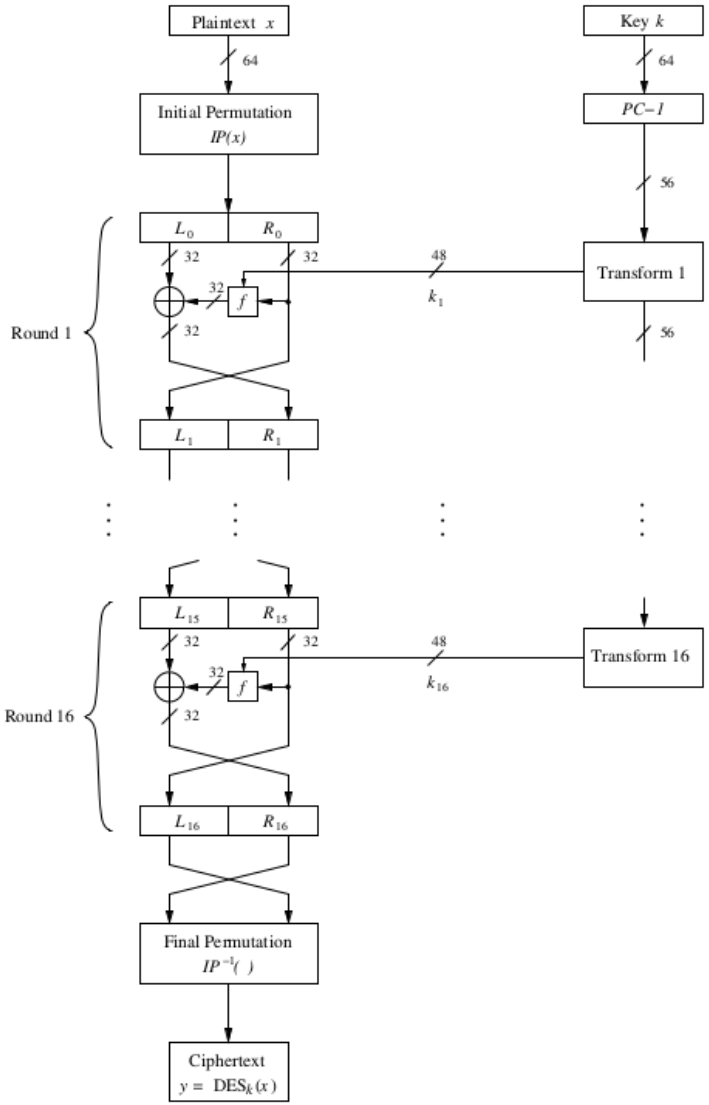


# La permutación inicial *IP*

## La permutación inicial *IP* es:

| *IP* | | | | | | | |
|---|---|---|---|---|---|---|---|
| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 |
| 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

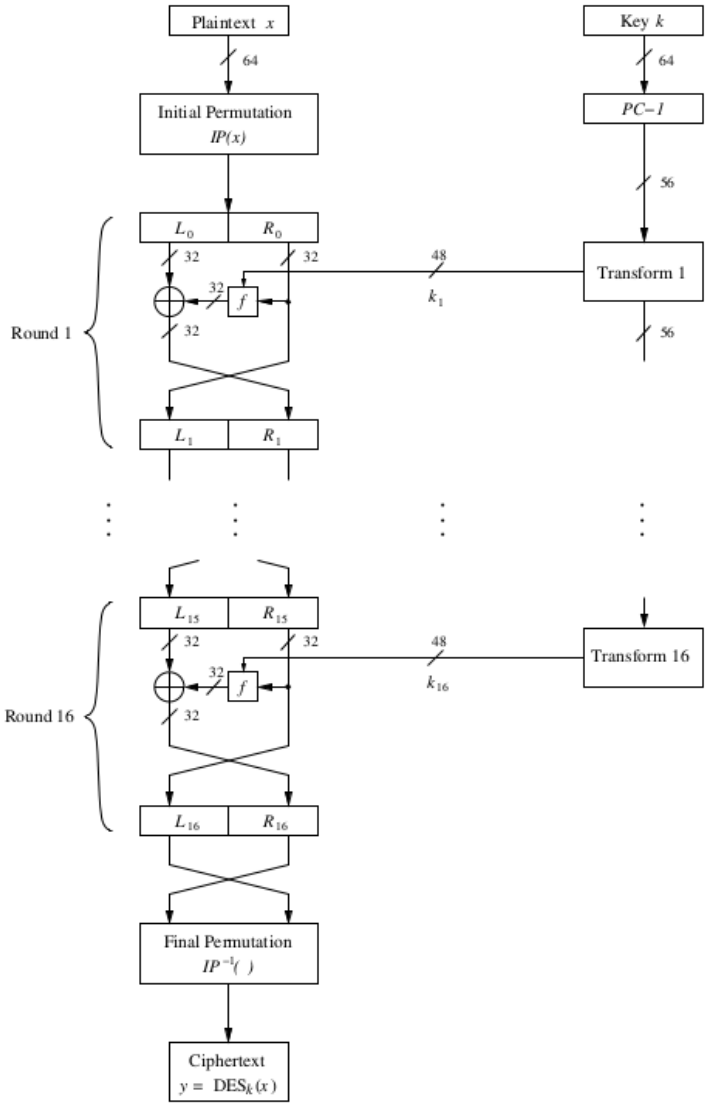
Esta permutación inicial *IP* se traduce de la siguiente manera: el bit de la posición 58 del plaintext $x$ es llevado a la primera posición en el vector de 64-bits $IP(x)$, la posición 50 del plaintext $x$ es llevado a la segunda posición de $IP(x)$ y así continua operando la permutación *IP* sobre el plaintext $x$.



# La permutación inicial *IP*

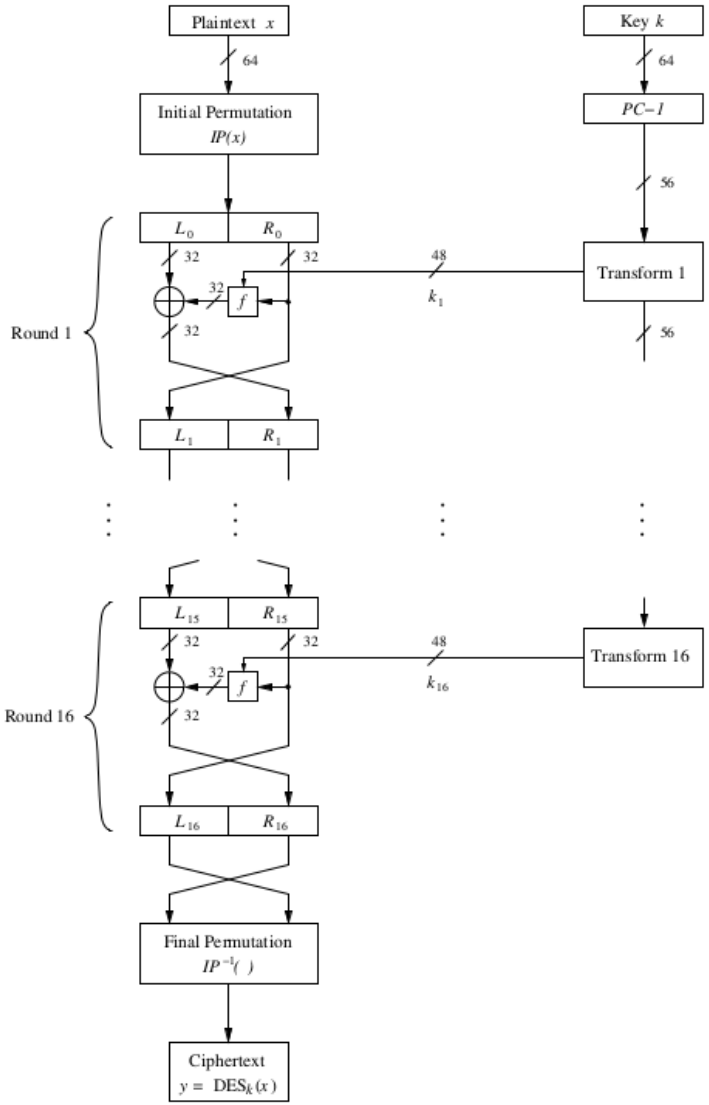
## La permutación inversa $IP^{-1}$ es:

| $IP^{-1}$ | | | | | | | |
|---|---|---|---|---|---|---|---|
| 40 | 8 | 48 | 16 | 56 | 24 | 64 | 32 |
| 39 | 7 | 47 | 15 | 55 | 23 | 63 | 31 |
| 38 | 6 | 46 | 14 | 54 | 22 | 62 | 30 |
| 37 | 5 | 45 | 13 | 53 | 21 | 61 | 29 |
| 36 | 4 | 44 | 12 | 52 | 20 | 60 | 28 |
| 35 | 3 | 43 | 11 | 51 | 19 | 59 | 27 |
| 34 | 2 | 42 | 10 | 50 | 18 | 58 | 26 |
| 33 | 1 | 41 | 9 | 49 | 17 | 57 | 25 |



# La función $f$

- La función $f$ juega una papel fundamental para la seguridad del DES.
- En la ronda $i$ esta función toma la mitad $R_{i-1}$ y la clave derivada $k_i$.
- La salida de la función $f$ es de 32-bits y usada con un XOR para encriptar la mitad izquierda $L_{i-1}$.



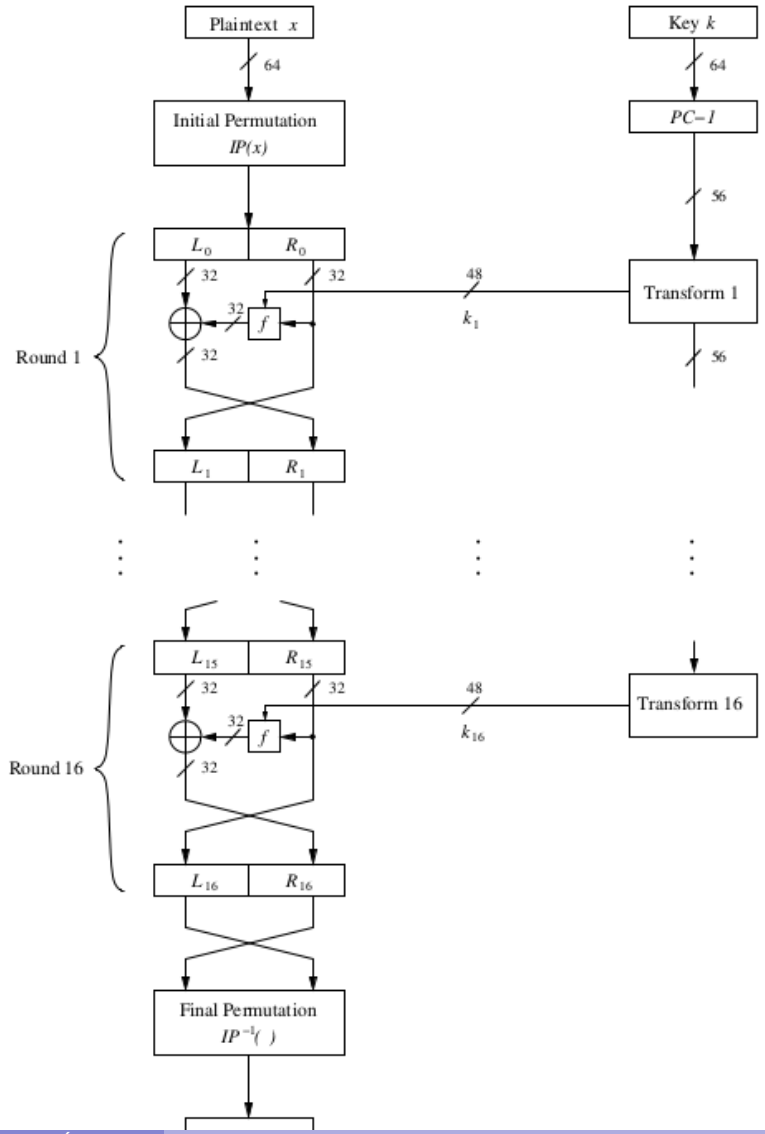
# La función $f$

Los siguientes pasos son ejecutados:

1. La mitad $R_{i-1}$ de largo de 32-bits es "expandida" a un vector de bits de largo de 48-bits de acuerdo a la *función de expansión* $E$. $E(R_{i-1})$ consiste de los 32 de $R_{i-1}$, permutado de cierta manera, con 16 bits aparecen dos veces, a través de la siguiente tabla de selección de bits:

| Función de expasión $E$ | | | | | |
|---|---|---|---|---|---|
| 32 | 1 | 2 | 3 | 4 | 5 |
| 4 | 5 | 6 | 7 | 8 | 9 |
| 8 | 9 | 10 | 11 | 12 | 13 |
| 12 | 13 | 14 | 15 | 16 | 17 |
| 16 | 17 | 18 | 19 | 20 | 21 |
| 20 | 21 | 22 | 23 | 24 | 25 |
| 24 | 25 | 26 | 27 | 28 | 29 |
| 28 | 29 | 30 | 31 | 32 | 1 |

2. El cálculo $E(R_{i-1})$ de 48-bits es XOR con la clave $k_i$ $(E(R_{i-1}) \oplus k_i)$ y el resultado es concatenado en en un vector de 8 posiciones de 6-bits cada uno, es decir, $B = B_1 B_2 B_3 B_4 B_5 B_6 B_7 B_8$.



# La función $f$

Los siguientes pasos son ejecutados:

1. Cada uno de estos subvectores $B_i$ de largo de 6-bits y son ingresados en 8 cajas de sustitución conocidas como $S-boxes$ y denotadas como $S_1, \ldots, S_8$, es decir, $S_i(B_i)$ para $i = 1, \ldots, 8$.
2. Estas cajas de sustitución $S_i$ son arreglos fijos de $4 \times 16$ cuyas entradas son valores enteros entre $0, \ldots, 15$.
3. Cada $B_i$ de largo 6-bits, digamos $B_i = b_1 b_2 b_3 b_4 b_5 b_6$, es transformado por la caja $S_i$ como $(S_i(B_i))$:
   - Los bits $b_1 b_6$ determina la representación binaria de la fila $r$ de $S_i$ $(0 \leq r \leq 3)$.
   - Los cuatro bits $b_2 b_3 b_4 b_5$ determina la representación binaria de la columna $c$ de $S_i$ $(0 \leq c \leq 15)$.
   - Entonces $S_i(B_i)$ se define como entradas $S_i(r, c)$ escritas en representación binaria que es un vector de bits de largo 4.

Ejemplo:

| $S_1$ | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 14 | 4 | 13 | 1 | 2 | 15 | 11 | 8 | 3 | 10 | 6 | 12 | 5 | 9 | 0 | 7 |
| 0 | 15 | 7 | 4 | 14 | 2 | 13 | 1 | 10 | 6 | 12 | 11 | 9 | 5 | 3 | 8 |
| 4 | 1 | 14 | 8 | 13 | 6 | 2 | 11 | 15 | 12 | 9 | 7 | 3 | 10 | 5 | 0 |
| 15 | 12 | 8 | 2 | 4 | 9 | 1 | 7 | 5 | 11 | 3 | 14 | 10 | 0 | 6 | 13 |

# La función $f$

- Esta salida $C_i = S_i(B_i)$ (con $1 \leq i \leq 8$) 4-bits para cada $C_i$.
- Luego el arreglo de bits $C = C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8$ tiene largo de 32-bits $(4 * 8)$ y se le aplica una permutación fija $P$.

| P | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 7 | 20 | 21 | 29 | 12 | 28 | 17 |
| 1 | 15 | 23 | 26 | 5 | 18 | 31 | 10 |
| 2 | 8 | 24 | 14 | 32 | 27 | 3 | 9 |
| 19 | 13 | 30 | 6 | 22 | 11 | 4 | 25 |

- Las cajas $S_i$ pueden ser revisadas en los libros:
  - Understanding Cryptography, A textbook for Student and Practitioners. Christof Paar and Jan Pelzl. Springer-Verlag Berlin Heidelberg 2010.
  - Cryptography Theory and Practice, Douglas R. Stinson, The CRC Press Series on Discrete Mathematics and Its Applications, 1995.

# La clave "Schedule"

La clave "Schedule" derivadas en las 16 rondas de ejecución del DES, se aplica un proceso que explicaremos en lo que sigue.

- La clave original es de longitud de 64-bits de los cuales 56-bits comprende la clave, ya que los bits de la clave en las posiciones $8, 16, \ldots, 64$ son eliminan de la clave y no incrementan la seguridad del DES, luego el DES es de 56-bit y no de 64-bit.
- En la siguiente figura, se puede ver que los primeros 64-bit de la clave son reducidos a 56 bits, ya que son ignorados a cada ocho bits.

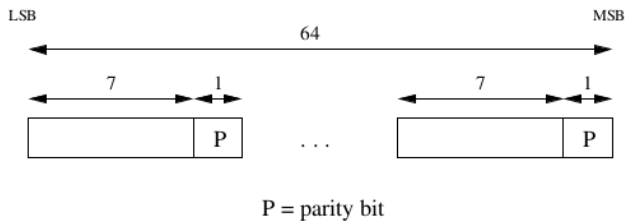



Figura: Clave de 64 bits de entrada y sus ocho bits de paridad

# La clave "Schedule"

- Estos 56 bits son permutados a través de PC-1. Se debe observar que no se permutan (ya que fueron eliminados) los bits de las posiciones $8, 16, \ldots, 64$.

| PC-1 | | | | | | |
|---|---|---|---|---|---|---|
| 57 | 49 | 41 | 33 | 25 | 17 | 9 |
| 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 10 | 2 | 59 | 51 | 43 | 35 | 27 |
| 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| 63 | 55 | 47 | 39 | 31 | 23 | 15 |
| 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 14 | 6 | 61 | 53 | 45 | 37 | 29 |
| 21 | 13 | 5 | 28 | 20 | 12 | 4 |

- El resultado de los 56-bits permutados es dividido en dos mitades $C_0$ y $D_0$, (donde los $C_0$ comprende los primeros 28-bits de la clave ya permutada PC-1(K) y $D_0$ los últimos 28 bits).

# La clave "Schedule"

- Para cada $i = 1, \ldots, 16$ calcular

$$C_i = LS_i(C_{i-1})$$
$$D_i = LS_i(D_{i-1}),$$

  $LS_i$ representa un "cyclic shift" a la izquierda de 1 o 2 posiciones, esto depende del valor de la iteración $i$ : shift en una posición si $i = 1, 2, 9,$ o 16 y un shift en dos posiciones en el otro caso.
- Luego se debe calcular $K_i$ =PC-2$(C_i D_i)$.
- Donde $PC - 2$ es otra permutación fija se debe observar que esta permutación elimina 8 bits de la clave $C_i D_i$, es decir, que la salida de esta permutación es de $48-$bits de la clave original $K$ y se define como:

| PC-2 | | | | | |
|---|---|---|---|---|---|
| 14 | 17 | 11 | 24 | 1 | 5 |
| 3 | 28 | 15 | 6 | 21 | 10 |
| 23 | 19 | 12 | 4 | 26 | 8 |
| 16 | 7 | 27 | 20 | 13 | 2 |
| 41 | 52 | 31 | 37 | 47 | 55 |
| 30 | 40 | 51 | 45 | 33 | 48 |
| 44 | 49 | 39 | 56 | 34 | 53 |
| 46 | 42 | 50 | 36 | 29 | 32 |

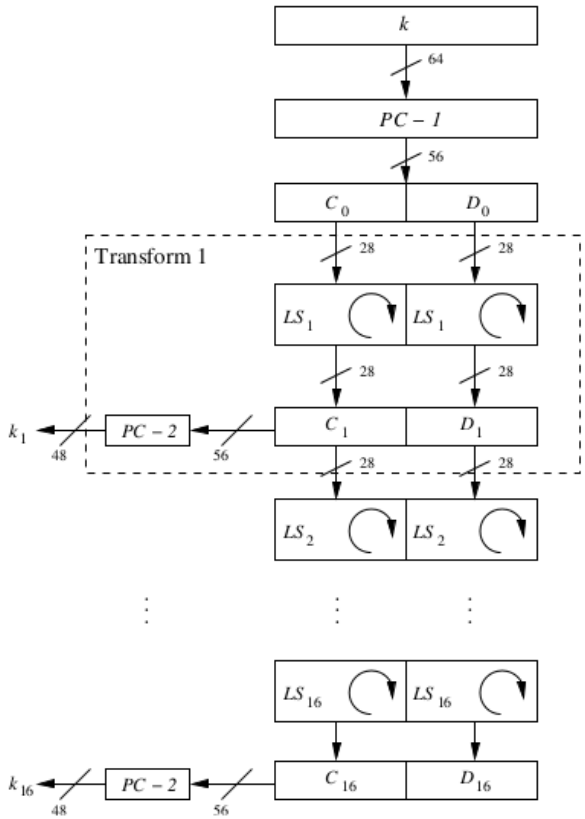



Figura: "Key Shedule" para el Algoritmo de Encriptación DES

El algoritmo para desencriptar el DES es el mismo que para encriptar, comenzando con el cibertext $y$ como entrada, pero usando el key schedule $k_{16}, \ldots, k_1$ en orden inverso. La salida es el plaintext $x$

# Seguridad Cifrado DES

- Uno de los problemas de este criptosistema es el tamaño del espacio de claves (keyspace) que es de $2^{56}$ considerado pequeño para aplicaciones reales y seguras.
- Varios maquinas de propósitos específicos han sido construidas para realizar un *plaintex attacks* este consiste esencialmente en una búsqueda exhaustiva de la clave utilizada. Es decir, dado el plaintext $x$ de 64-bits y le corresponde un cibertext $y$ la idea es utilizar las diferentes claves $K$ tal que $e_K(x) = y$ sea encontrada se debe observar que solo existe una sola clave $K$ que cumple $e_K(x) = y$.
- En CRYPTO 93 Rump Session, Michel Wiener presento una detallada maquina de búsqueda de claves. La maquina se basa en la búsqueda de la clave utilizando pipelined, que realiza 16 encriptaciones simultáneamente. Este chip puede testar $5 \times 10^7$ claves por segundo.

# Seguridad Cifrado DES

- Uno de los problemas de este criptosistema es el tamaño del espacio de claves (keyspace) que es de $2^{56}$ considerado pequeño para aplicaciones reales y seguras.
- Varios maquinas de propósitos específicos han sido construidas para realizar un *plaintex attacks* este consiste esencialmente en una búsqueda exhaustiva de la clave utilizada. Es decir, dado el plaintext $x$ de 64-bits y le corresponde un cibertext $y$ la idea es utilizar las diferentes claves $K$ tal que $e_K(x) = y$ sea encontrada se debe observar que solo existe una sola clave $K$ que cumple $e_K(x) = y$.
- En CRYPTO 93 Rump Session, Michel Wiener presento una detallada maquina de búsqueda de claves. La maquina se basa en la búsqueda de la clave utilizando pipelined, que realiza 16 encriptaciones simultáneamente. Este chip puede testar $5 \times 10^7$ claves por segundo.

# Seguridad Cifrado DES

- Uno de los problemas de este criptosistema es el tamaño del espacio de claves (keyspace) que es de $2^{56}$ considerado pequeño para aplicaciones reales y seguras.
- Varios maquinas de propósitos específicos han sido construidas para realizar un *plaintex attacks* este consiste esencialmente en una búsqueda exhaustiva de la clave utilizada. Es decir, dado el plaintext $x$ de 64-bits y le corresponde un cibertext $y$ la idea es utilizar las diferentes claves $K$ tal que $e_K(x) = y$ sea encontrada se debe observar que solo existe una sola clave $K$ que cumple $e_K(x) = y$.
- En CRYPTO 93 Rump Session, Michel Wiener presento una detallada maquina de búsqueda de claves. La maquina se basa en la búsqueda de la clave utilizando pipelined, que realiza 16 encriptaciones simultáneamente. Este chip puede testar $5 \times 10^7$ claves por segundo.